UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/887,576	06/25/2001	Irit Loy	LOY=3	7562

1444 7590 05/31/2005

BROWDY AND NEIMARK, P.L.L.C.
624 NINTH STREET, NW
SUITE 300
WASHINGTON, DC 20001-5303

EXAMINER	
LU, KUEN S	
ART UNIT	PAPER NUMBER
2167	

DATE MAILED: 05/31/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/887,576	**Applicant(s)** LOY ET AL.
	Examiner Kuen S Lu	**Art Unit** 2167

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 March 2005*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-69* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *1-69* is/are allowed.

6)□ Claim(s) _____ is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

Response to Amendment

1. The Applicant's Amendments, filed on March 15, 2005, is noted and considered. As per Applicant's Remarks, please refer to the section ***Remarks*** after the Office Action for non-Final Rejection (hereafter "the Action") under U.S.C. 35 §103(a), as stated below.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C.103(a).

3. Claims 1-2, 17-19, 23-24, 39-41, 45-46, 61-63 and 67-69 are rejected under 35 U.S.C. 103(a) as being unpatentable over SunHPC (Sun HPC ClusterTools 3.1, Administrator's Guide, Sun Microsystem, March 2000, hereafter "SunHPC").

As per claim 1, 23 and 45, SunHPC teaches the following:

"a cluster of computing nodes having shared access to one or more file system in data storage using parallel file system software" (See Page 45 and Fig. 4-2 wherein

SunHPC's cluster shares two parallel file systems is equivalent to Applicant's a cluster of computing nodes having shared access to one or more file system in data storage using parallel file system software);

"initiating a session of a data management application on a first one of the nodes" (See Page 47 wherein SunHPC's when an application issues a file system-related system call, the VFS layer identifies the particular file system involved, and pass the request on to the responsible kernel module and further, when Sun PFS kernel is passed a file system request, will likely marshal the request with parameters and passes to a PFS proxy daemon is equivalent to Applicant's initiating a session of a data management application on a first one of the nodes); and

"receiving a request submitted to the parallel file system software at the second one of the nodes to mount one of the file systems in the data storage on the second one of the nodes" (See Pages 45 and 53 wherein SunHPC's execution of mount with qualifier pfs or pfsmount command to mount PFS from a client having a PFS proxy daemon or a single node to cause PFS file system mounted on all nodes in the cluster or a cluster partition is equivalent to receiving a request submitted to the parallel file system software at the second one of the nodes to mount one of the file systems in the data storage on the second one of the nodes).

SunHPC does not explicitly teach "sending a mount event message from the second node to the first node responsive to the request, for processing by the data management application on the first node", although SunHPC teaches when a single node mounts a PFS file system and the file system is mounted to other cluster nodes and PFS proxy daemon service on the cluster nodes, as previously described.

However, SunHPC teaches CRE (cluster run-time environment) to support MPI (message passing interface) protocol to stripe messages over cluster network (See Pages 3, 8 and 34), verifying PFS is correctly mounted before using the file system and logging daemon messages to store into file (See Pages 53 and 128), and further teaches encountering users with a permission denied message when users access a not visible file system.

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine SunHPC's teachings on sending access denial message over the cluster nodes and logging daemon messages by broadcasting PFS successfully mounted message to the nodes on the cluster or cluster partition because the combined teaching would have enabled all nodes of SunHPC system to share data stores immediately after it is mounted and eliminate the need to further verify if the file system is correctly mount via performing additional proxy daemon work such that I/O performance would have been be further improved on the normally would have been an excessively inefficient and complex clustering parallel file system.

As per claims 2, 24 and 46, SunHPC further teaches the "mounting first and second instances of the one of the file systems on the first and second nodes, respectively, responsive to the mount event message" (See Pages 3, 8, 34, 45, 47, 53 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and a node mounts PFS for usage by other nodes, for example, pfs-demo0 and pfs-demo1 are respectively mounted on two set of nodes is equivalent to Applicant's mounting first and second instances of the one of the file systems on the first and second nodes, respectively, responsive to the mount event message).

As per claims 17, 39 and 61, SunHPC further teaches the following:
"receiving a response to the mount event message from the data management application on the first node" (See Pages 3, 8, 34, 45, 47, 53 and 128 wherein SunHPC's when an application issues a file system-related system call, the VFS layer identifies the particular file system involved, and pass the request on to the responsible kernel module and further, when Sun PFS kernel is passed a file system request, will likely marshal the request with parameters and passes to a PFS proxy daemon service which is available on cluster nodes and message is passing between nodes, mounts parallel file systems and issues file system-related system call is equivalent to Applicant's receiving a response to the mount event message from the data management application on the first node); and
"mounting an instance of the one of the file systems on the second node subject to the response from the data management application on the first node" (See Pages 3, 8, 34, 45, 47, 53 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and a node mounts PFS for usage by other nodes, for example, pfs-demo0 and pfs-demo1 are respectively mounted on two set of nodes is equivalent to Applicant's mounting an instance of the one of the file systems on the

second node subject to the response from the data management application on the first node).

As per claims 18, 40 and 62, SunHPC further teaches the following:

"receiving a further request submitted to the parallel file system software to mount the one of the file systems on a further one of the nodes" (See Pages 45 and 53 wherein SunHPC's execution of mount with qualifier pfs or pfsmount command to mount PFS from a client having a PFS proxy daemon or a single node of the cluster to cause PFS file system mounted on all nodes in the cluster or a cluster partition is equivalent to Applicant's receiving a further request submitted to the parallel file system software to mount the one of the file systems on a further one of the nodes); and

"sending a further mount event message responsive to the further request for processing by the data management application on the first node" (See Pages 3, 8, 34, 45, 47, 53 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and a node mounts PFS for usage by other nodes, for example, pfs-demo0 and pfs-demo1 are respectively mounted on two set of nodes, and note the mount process is equivalent to Applicant's sending a further mount event message responsive to the further request for processing by the data management application on the first node).

As per claims 19, 41 and 63, SunHPC further teaches "the further one of the nodes is the first node" (See Pages 45 and 53 wherein SunHPC's execution of mount with qualifier pfs or pfsmount command to mount PFS from a client having a PFS proxy daemon or a single node of the cluster without restriction to a specific node to cause PFS file system mounted on all nodes in the cluster or a cluster partition is equivalent to Applicant's the further one of the nodes is the first node).

As per claims 67, 68 and 69, SunHPC further teaches "request to mount one of the file systems is submitted by a user application running on the second one of the nodes" (See Page 47 wherein SunHPC's when an application issues a file system-related system call, the VFS layer identifies the particular file system involved, and pass the request on to the responsible kernel module and further, when Sun PFS kernel is passed a file system request, will likely marshal the request with parameters and passes to a PFS proxy daemon is equivalent to Applicant's request to mount one of the file systems is submitted by a user application running on the second one of the nodes).

4. Claims 3-8, 25-30 and 47-52 are rejected under 35 U.S.C. 103(a) as being unpatentable over SunHPC (Sun HPC ClusterTools 3.1, Administrator's Guide, Sun Microsystem, March 2000, hereafter "SunHPC") as applied to claims 1-2, 23-24 and 45-46, and further in view of SunSoft (NFS File Sets, Connectathon '97, Eisler et al., 1997, SunSoft, hereafter "SunSoft").

As per claims 3, 25 and 47, SunHPC further teaches "receiving a further request at the second node to unmount the second instance of the one of the file systems at the second node" (See Pages 3, 8, 34, 45, 47, 53 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and further at Pages 54-55 where SunHPC command umount or pfsumount is executed to unmount PFS from one or multiple nodes is equivalent to Applicant's receiving a further request at the second node to unmount the second instance of the one of the file systems at the second node).

The SunHPC reference does not specifically teach "sending, responsive to the further request, a preunmount event message to the first node", although the reference teaches "sending, responsive to the further request, an unmount event message to the first node" (See Pages 3, 8, 34, 45, 47, 53 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and further at Pages 54-55 where SunHPC command umount or pfsumount is executed to unmount PFS from one or multiple nodes).

However, SunSoft teaches premounting of shared library file system in the NFS file sets (See Page 25).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine SunSoft's teaching with the SunHPC reference because all references teach shared file system and mounting file system with proper knowledge, the further combination of the teachings would have enabled SunHPC' system properly mounting and unmounting file system with proper parameter settings for operating file system more flexibly and efficiently.

The combined teaching of SunSoft and SunHPC references further teaches "responding to the preunmount event message so as to permit unmounting of the second file system instance without unmounting the first file system instance" (See SunSoft: Page 25 where preunmounting of shared library file system is explicitly taught, SunHPC: Pages 3, 8, 34, 45, 47, 53 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and further at Pages 54-55 wherein SunHPC' each node has own mounting point for unmounting a parallel file system without unmounting other file systems is equivalent to Applicant's responding to the preunmount event message so as to permit unmounting of the second file system instance without unmounting the first file system instance).

As per claims 4, 26 and 48, the combined teaching of SunSoft and SunHPC references further teaches "responding to the preunmount event message comprises determining at the first node, responsive to one or more flags set in the preunmount event message, whether the request was submitted on the first node or on another one of the nodes" (Pages 3, 8, 34, 45, 47, 53 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and further at Pages 54-55 where a PFS can be unmounted from an individual node or all nodes of a cluster, and SunSoft: Pages 23-25 teaching preunmounting shared library file system is equivalent to Applicant's responding to the preunmount event message comprises determining at the first node, responsive to one or more flags set in the preunmount event message, whether the request was submitted on the first node or on another one of the nodes).

As per claims 5, 27 and 49, the combined teaching of SunSoft and SunHPC references teaches the following:

"receiving the preunmount event message at the first node" (See SunHPC: Pages 3, 8, 34, 45, 47, 53 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and further at Pages 44-45 where parallel file system is operated on a cluster of a plurality of computer nodes, and SunSoft: Pages 23-25 teaches preunmounting and premounting shared file system is equivalent to Applicant's receiving the preunmount event message at the first node);

"obtaining a data management access right from a physical file system (PFS) software component at the first node responsive to the preunmount event message" (See SunHPC: Pages 44-45 where parallel file system is operated on a cluster of a plurality of computer nodes and further at Page 47 where an application issues a file system-related system call, the VFS layer identifies and passes the request to the kernel module of the responsible file system, and where the module marshals the request arguments and further passes the request to proxy daemon , SunSoft: Pages 23-25 teaches preunmounting and premounting shared file system is equivalent to Applicant's obtaining a data management access right from a physical file system (PFS) software component at the first node responsive to the preunmount event message); and "processing the preunmount event message using the access right" (See SunHPC: Pages 44-45 and 47 where parallel file system is operated on a cluster of a plurality of computer nodes and an application issues a file system-related system call, the VFS layer identifies and passes the request to the kernel module of the responsible file system, and where the module marshals the request arguments and further passes the request to proxy daemon, and SunSoft: Pages 23-25 teaches preunmounting and premounting shared file system, and is equivalent to Applicant's processing the preunmount event message using the access right).

As per claims 6, 28 and 50, the combined teaching of SunSoft and SunHPC references teaches the following:

"receiving the request comprises receiving first and second requests to mount different ones of the file systems in the data storage" (See SunHPC: Pages 3, 8, 34, 45, 47, 53 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and further at Pages 45, 47 and 53 where parallel file systems are mounted on multiple nodes is equivalent to Applicant's receiving the request comprises receiving first and second requests to mount different ones of the file systems in the data storage);

"wherein receiving the further request comprises receiving further first and second requests to unmount the different ones of the file systems" (See SunHPC: Pages 3, 8, 34, 45, 47, 53 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and further at Pages 45 and 54-55 where parallel file systems are mounted on multiple nodes and unmount individually or together is equivalent to Applicant's wherein receiving the further request comprises receiving further first and second requests to unmount the different ones of the file systems), and

"the preunmount event message comprises, responsive to dispositions set for the different ones of the file systems" (See SunHPC: Pages 3, 8, 34, 45, 47, 53 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and further at Page 45 where a plurality of parallel file systems are operated, and SunSoft: Pages 23-25 where shared library file systems are preunmounted is equivalent to Applicant's the preunmount event message comprises, responsive to dispositions set for the different ones of the file systems); and

"sending a first preunmount event message to the first node responsive to the first unmount request and sending a second preunmount event message to the second unmount request to a further node, on which a further data management application session has been initiated" (See SunHPC: Pages 3, 8, 34, 47 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and further at Pages 45, 53-55 where a plurality of parallel file systems are mounted, operated and unmounted individually or together, and SunSoft: Pages 23-25 teaches coding mounting, unmounting and their preoperation of files systems into logical grouping wherein SunHPC' is equivalent to Applicant's sending a first preunmount event message to the first node responsive to the first unmount request and sending a second preunmount event message to the second unmount request to a further node, on which a further data management application session has been initiated).

As per claims 7, 29 and 51, the combined teaching of SunSoft and SunHPC references further teaches "responding to the preunmount event message comprises sending a reply to the message from the first node to the second node, and comprising, responsive to the reply, unmounting the second file system instance and sending an unmount event message from the second node to the first node" (See SunHPC: Pages 3, 8, 34, 45, 47 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and further at Pages 53-55 where parallel file systems are mounted, operated and unmounted individually or together, and SunSoft: Pages 23-25 teaches coding mounting, unmounting and their preoperation of files systems into logical grouping is equivalent to Applicant's responding to the preunmount event message comprises sending a reply to the message from the first node to the second node, and comprising, responsive to the reply, unmounting the second file system instance and sending an unmount event message from the second node to the first node).

As per claims 8, 30 and 52, the combined teaching of SunSoft and SunHPC references further teaches "determining at the first node, responsive to one or more flags set in the unmount event message, whether the further request was submitted on the first node or on another one of the nodes" (See SunHPC: Pages 3, 8, 34, 45, 47, 53 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and further at Pages 53-55 where parallel file systems are mounted, operated and unmounted individually or together, and SunSoft: Pages 23-25 teaches " Knowledge"' on unmounting files systems is equivalent to Applicant's determining at the first node, responsive to one or more flags set in the unmount event message, whether the further request was submitted on the first node or on another one of the nodes).

5. Claims 9, 31 and 53 are rejected under 35 U.S.C. 103(a) as being unpatentable over SunHPC (Sun HPC ClusterTools 3.1, Administrator's Guide, Sun Microsystem, March 2000, hereafter "SunHPC") as applied to claims 1, 23 and 45, and further in view of SunSoft (NFS File Sets, Connectathon '97, Eisler et al., 1997, SunSoft, hereafter "SunSoft").

As per claims 9, 31 and 53, SunHPC reference teaches "determining at the first node"..."whether the further request was submitted on the first node or on another one of the nodes" (See SunHPC: Pages 3, 8, 34, 45, 47 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and further at Pages 53-55 where parallel file systems are mounted, operated and unmounted individually or together).

SunHPC does not specifically teach "responsive to one or more flags set in the mount event message" for "determining at the first node, responsive to one or more flags set in the mount event message, whether the further request was submitted on the first node or on another one of the nodes".

However, SunSoft teaches one or more flags for the mount event message (See Pages 23-25 where mounting flags such as postmount, premount and preunmount are coded).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine SunSoft's teaching with SunHPC reference because all references teach shared file system and mounting file system with proper knowledge, the further combination of the teachings would have enabled SunHPC' system properly mounting and unmounting file system with proper parameter settings for operating file system more flexibly and efficiently.

6. Claims 20-21, 42-43, and 64-65 are rejected under 35 U.S.C. 103(a) as being unpatentable over SunHPC (Sun HPC ClusterTools 3.1, Administrator's Guide, Sun Microsystem, March 2000, hereafter "SunHPC") as applied to claims 1, 18-19, 23, 40-41, 45 and 62-63, and further in view of SunSoft (NFS File Sets, Connectathon '97, Eisler et al., 1997, SunSoft, hereafter "SunSoft").

As per claims 20, 42 and 64, SunHPC teaches "receiving first and second unmount requests to unmount the file system from the second node and from the further one of the nodes" (See SunHPC: Pages 3, 8, 34, 45, 47 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and further at Pages 53-55 where parallel file systems are mounted, operated and unmounted individually or together).

SunHPC does not specifically teach "generating first and second preunmount event messages", although the combined reference teaches "at the second node and at the further one of the nodes responsive to the first and second unmount requests, for processing by the data management application on the first node" (See SunHPC: Pages 3, 8, 34, 45, 47 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and further at Pages 53-55 where parallel file systems are mounted, operated and unmounted individually or together).

However, SunSoft teaches coding mounting, unmounting, preunmounting and premounting of files systems at Pages 23-25.

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine SunSoft's teaching with SunHPC reference because all references teach shared file system and mounting file system with proper knowledge, the further combination of the teachings would have enabled SunHPC' system properly mounting and unmounting file system with proper parameter settings for operating file system more flexibly and efficiently.

As per claims 21, 43 and 65, the combined teaching of Sunsoft and SunHPC references further teaches "sending a reply to the first and second preunmount event messages from the first node to the second node and to the further one of the nodes, and, responsive to the reply, unmounting the file system at the second node and the further one of the nodes, and generating respective unmount event messages at the second node and at the further one of the nodes" (See SunHPC: Pages 3, 8, 34, 45, 47 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and further at Pages 53-55 where parallel file systems are mounted, operated and unmounted individually or together, and SunSoft: Pages 23-25 teaches coding mounting, unmounting and their preoperation of files systems into logical grouping is equivalent to Applicant's sending a reply to the first and second preunmount event messages from the first node to the second node and to the further one of the nodes, and, responsive to the reply, unmounting the file system at the second node and the further one of the nodes, and generating respective unmount event messages at the second node and at the further one of the nodes).

7. Claims 10, 13-16, 22, 32, 35-38, 44, 54, 57-60 and 66 are rejected under 35 U.S.C. 103(a) as being unpatentable over SunHPC (Sun HPC ClusterTools 3.1, Administrator's Guide, Sun Microsystem, March 2000, hereafter "SunHPC") as applied to claims 1, 23 and 45 above, and further in view of Bober (U.S. Patent 6,718,372).

As per claims 10, 32 and 54, SunHPC teaches "parallel file system" (See Page 45 and Fig. 4-2 wherein SunHPC's cluster shares two parallel file systems is equivalent to Applicant's a cluster of computing nodes having shared access to one or more file system in data storage using parallel file system software) and "initiating a session" of the parallel file system software (See Page 47 wherein SunHPC's when an application issues a file system-related system call, the VFS layer identifies the particular file system involved, and pass the request on to the responsible kernel module).

SunHPC does not specifically teach initiating a session "in accordance with a data management application programming interface" of the parallel file system software.

However, Bober teaches "initiating a session of a data management application interface" (See Figs. 6-7A and col. 21, line 66 – col. 22, line 21 wherein Bober's different APIs interface shared data storage and data management is equivalent to Applicant's initiating a session of a data management application).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine Bober's teaching with the SunHPC reference because all references are devoted to shared file system application on computer cluster platform and the further combined reference would have enabled SunHPC system to support data management in a parallel file system environment by using a resource management application interface to create highly available data services for efficiently utilizing computer cluster resources.

The combined teaching of Bober and SunHPC references further teaches "wherein receiving the request and sending the mount event message using the DMAPI" (See Bober: Figs. 6-7A and col. 21, line 66 – col. 22, line 21 where application programming interface is utilized for data management functions, and SunHPC: Pages 3, 8, 34, 45, 47, 53 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and a node mounts PFS for usage by other nodes, further at Pages 11-12 where parallel file system is mounted).

As per claims 13, 35 and 57, the combined teaching of Bober and SunHPC references further teaches "receiving and processing the event message at the first node using one or more functions of the DMAPI called by the data management application" (See Bober: Figs. 6-7A and col. 21, line 66 – col. 22, line 21 where application programming interface is utilized for data management functions while SunHPC teaches PFS proxy daemon service available on cluster nodes and message passing between nodes at See SunHPC: Pages 3, 8, 34, 45, 47 and 128).

As per claims 14, 36 and 58, the combined teaching of Bober and SunHPC references further teaches "sending the event message comprises sending the messages for processing in accordance with a disposition specified by the data management application using the DMAPI for association with an event generated by the operation" (See SunHPC: Pages 3, 8, 34, 45, 47 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and Bober: Figs. 6-7A and col. 21, line 66 – col. 22, line 21 where application programming interface is utilized for data management functions).

As per claims15, 37 and 59, the combined teaching of Bober and SunHPC references further teaches "sending event message comprises setting one or more flags in the message to indicate whether the request was submitted on the first node or on another one of the nodes" (See SunHPC: Pages 3, 8, 34, 45, 47 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes).

As per claims 16, 38 and 60, the combined teaching of Bober and SunHPC references further teaches "invoking a function of the DMAPI to obtain mount information regarding the one of the file systems, and wherein a response provided by the function, one or more flags are set to indicate whether the one of the file systems is mounted on the first node or another one of the nodes in the cluster or on both the first node and on another one of the nodes in the cluster" (See Bober: Figs. 6-7A and col. 21, line 66 – col. 22, line 21 where application programming interface is utilized for data management functions and SunHPC: Fig. 1, Pages 4 and 10-11 where parallel file system is mounted with specific mount setting for the mounting directory and device files).

As per claims 22, 44 and 66, the combined teaching of Bober and SunHPC references further

teaches "initiating the session of the data management application comprises initiating a data migration application, so as to free storage space on at least one of the volumes of data storage" (See Bober: Figs. 6-7A and col. 21, line 66 – col. 22, line 21 where application programming interface is utilized for data management functions is equivalent to Applicant's initiating the session of the data management application comprises initiating a data migration application, so as to free storage space on at least one of the volumes of data storage).

8. Claims 11-12, 33-34 and 55-56 are rejected under 35 U.S.C. 103(a) as being unpatentable over SunHPC (Sun HPC ClusterTools 3.1, Administrator's Guide, Sun Microsystem, March 2000, hereafter "SunHPC") in view of Bober (U.S. Patent 6,718,372), as applied to claims 10, 32 and 54 above, and further in view of SunSoft (NFS File Sets, Connectathon '97, Eisler et al., 1997, SunSoft, hereafter "SunSoft").

As per claims 11, 33 and 55, the combined teaching of Bober and SunHPC references further teaches "receiving an unmount request to unmount the file system from the second node using DMAPI, and sending" an event message to the first node responsive to the unmount request using the DMAPI, for processing by the data management application on the first node" (See Bober: Figs. 6-7A and col. 21, line 66 – col. 22, line 21 where application programming interface is utilized for data management functions, and See SunHPC: Pages 3, 8, 34, 45, 47 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and further at Fig. 1, Pages 4 and 15-16 where parallel file system is unmounted).

The combined reference does not specifically teach "sending a preunmount event message" to the first node responsive to the unmount request.

However, SunSoft teaches coding mounting, unmounting, preunmounting and premounting of files systems at Pages 23-25.

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine SunSoft's teaching with the Bober and SunHPC references because the references teach shared file system and mounting file system with proper knowledge, the further combination of the teachings would have enabled SunHPC' system properly mounting and unmounting file system with corresponding flags or settings, via message queue among cluster nodes for operating file system more flexibly and efficiently.

As per claims 12, 34 and 56, the combined teaching of SunSoft, Bober and SunHPC references further teaches "responding to the preunmount event message comprises sending a reply to the message from the first node to the second node, and comprising, responsive to the reply, unmounting the second file system instance and sending an unmount event message from the second node to the first node" (See Bober: Figs. 6-7A and col. 21, line 66 – col. 22, line 21 where application programming interface is utilized for data management functions, See SunHPC: Pages 3, 8, 34, 45, 47 and 128 where PFS proxy daemon service is available on cluster nodes and message is passing between nodes, and further at Fig. 1, Pages 4 and 15-16 where parallel file system is unmounted, and SunSoft: Pages 23-25 mounting, unmounting, preunmounting and premounting of files systems are coded).

Remarks

9. The Applicants' amendments filed on March 15, 2005 are noted. The Applicant's Remarks concerning Miller reference, assigned to and co-owned by IBM, being applied for rejecting claims 1-69 were found convincing and considered. However, after a further search, the Examiner has identified and further combined the teachings from the applied prior art.

Conclusion

10. The prior art made of record

H. U.S. Patent No. 6,718,372

V. NFS File Sets, Connectathon '97, Eisler et al., 1997, SunSoft

X. Sun HPC ClusterTools 3.1, Administrator's Guide, Sun Microsystem, March 2000

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U. Using the Parallel Virtual File System, Ross et al., Clemson University, January 2001

W. Sun Cluster 3.0 Data Services Developers' Guide, November, 2000, Sun Microsystems, Inc.

A.	U.S. Patent No.	6,625,639
B.	U.S. Patent No.	6,363,411
C.	U.S. Patent No.	6,192,408
D.	U.S. Patent No.	6,275,953
E.	U.S. Patent No.	6,151,684
F.	U.S. Patent No.	6,202,080
G.	U.S. Patent No.	6,249,879

Contact information

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kuen S. Lu whose telephone number is (571) 272-4114. The examiner can normally be reached on Monday-Friday (8:30 am-5:30 pm). If attempts to reach the examiner by telephone pre unsuccessful, the examiner's supervisor, John E Breene can be reached on (571) 272-4107. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for Page 13 published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Kuen S. Lu	Mohammad Ali
Patent Examiner	Primary Examiner
May 23, 2005	May 23, 2005